

AMENDMENTS TO THE CLAIMS:

Please amend claim 53 as indicated in the below listing of claims. This listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Presented) A computer-implemented method for inlining code of a computer program, comprising:

identifying a subprogram of the computer program, wherein the subprogram exhibits varying execution characteristics associated with corresponding execution paths;

identifying a range of variables associated with a first execution characteristic of the subprogram;

associating the range of variables with a first execution path of the subprogram;

coding an inline directive as part of a program comment statement associated with the first execution path; and

selectively inlining computer code of the first execution path of the subprogram based on the inline directive.

2. (Previously Presented) The method of claim 1, wherein the varying execution characteristics are each associated with certain ranges of variables.

3. (Previously Presented) The method of claim 1 wherein selectively inlining includes:

leaving intact a second execution path that exhibits a second execution characteristic different from the first characteristic based on a second range of variables.

4. (Previously Presented) The method of claim 1, wherein the execution characteristics are based on an execution time for the execution paths.

5. (Original) The method of claim 4 wherein the execution characteristics are based on the frequency of execution of the paths.

Claims 6-9. (Previously Canceled)

10. (Previously Presented) A computer program compiler for inlining computer program code, comprising:

a subprogram identification module that identifies subprograms of the computer code by

identifying ranges of variables that each relate to a corresponding execution characteristic of a corresponding subprogram included in the computer code,

associating the ranges of variables with certain execution paths of each of the subprograms, and

inserting inline directives in a part of a program comment statement associated with each of the certain execution paths; and

a path identification module that selectively inlines computer code of the certain execution paths of the subprograms based on the inline directives.

Claims 11-13 (Previously Canceled).

14. (Previously Presented) A computer-implemented method of determining whether to replace subprogram code of a computer program, comprising the steps of:

identifying a subprogram that has a first and a second execution characteristic;
identifying certain arguments that cause the subprogram to exhibit the first execution characteristic;

associating the certain arguments that cause the subprogram to exhibit the first execution characteristic with a portion of the subprogram;

replacing the portion of the subprogram that exhibits the first execution characteristic with program instructions that explicitly define operations of the first execution characteristic; and

leaving intact a second portion of the subprogram that exhibits the second execution characteristic.

15. (Previously Presented) The method of claim 14 wherein said second execution characteristic is an atypical characteristic associated with program code that requires special processing.

16. (Original) The method of claim 15 wherein said atypical characteristic is an execution time duration.

17. (Original) The method of claim 16 wherein said execution time duration exceeds a predetermined threshold.

18. (Previously Presented) The method of claim 17 wherein said first execution characteristic is a typical execution characteristic associated with program code that requires normal processing.

19. (Original) The method of claim 17 wherein said first and second execution characteristics are execution time durations.

Claim 20. (Previously Canceled)

21. (Previously Presented) The method of claim 14 wherein the certain arguments are associated with conditional execution computer statements associated with the first characteristic.

22. (Previously Presented) The method of claim 19 wherein the first and second execution characteristics are invoked based on conditional execution computer statements associated with the characteristics.

23. (Previously Presented) A computer readable medium for inlining computer program code, which when executed by a computer, performs the steps of:

identifying a subprogram that has a plurality of execution characteristics by identifying ranges of variables that cause the subprogram to exhibit the execution characteristics, associating a certain range of variables with one of the execution characteristics, and associating the certain range of variables with a selected portion of the subprogram that corresponds to the one execution characteristic;

inlining only the selected portion of the subprogram that corresponds to the one execution characteristic,

wherein the selected portion is defined by a selected path of a plurality of execution paths that may be executed by the subprogram and the selected path is determined by identifying a non-executable statement configured to direct the computer to interpret at least a portion of the non-executable statement as a special directive.

Claims 24 and 25. (Previously Canceled)

26. (Previously Presented) The medium of claim 23 wherein the program comment statement is included in the selected path.

27. (Original) The medium of claim 23 wherein the execution characteristics are identified by evaluating a conditional execution statement associated with a subprogram call.

28. (Previously Presented) A computer-implemented method of replacing subprogram code in a computer system, comprising the steps of:

identifying a subprogram that operates in a first manner when operands passed to the subprogram fall within a first range of values and that operates in a second manner when operands passed to the subprogram fall within a second range of values;

associating the operands with the first range of values with an execution path of the subprogram;

replacing subprogram statements included in the execution path that cause the subprogram to operate in the first manner with expanded code.

Claims 29-34. (Previously Canceled)

35. (Previously Presented) An apparatus having a processor and a memory containing programs for inlining code of a computer program which when executed using the processor perform steps comprising:

identifying a subprogram of the computer program, wherein the subprogram exhibits varying execution characteristics associated with corresponding execution paths, the identifying including:

identifying a range of variables associated with a first execution characteristic of the subprogram,

associating the range of variables with a first execution path of the subprogram, and

coding an inline directive as part of a program comment statement associated with the first execution path; and selectively inlining computer code of the first execution path of the subprogram based on the inline directive.

36. (Previously Presented) The apparatus of claim 35, wherein the varying execution characteristics are each associated with certain ranges of variables.

37. (Previously Presented) The apparatus of claim 35 wherein selectively inlining includes:

leaving intact a second execution path that exhibits a second execution characteristic different from the first characteristic based on a second range of variables.

38. (Previously Presented) The apparatus of claim 37 wherein the execution characteristics are based on an execution time for the execution paths.

39. (Original) The apparatus of claim 37 wherein the execution characteristics are based on the frequency of execution of the paths.

Claims 40-43. (Previously Canceled)

44. (Previously Presented) An apparatus having a processor and a memory containing programs for determining whether to replace subprogram code of a computer program which when executed using the processor perform steps comprising:

identifying a subprogram that has a first and a second execution characteristic;
identifying certain arguments that cause the subprogram to exhibit the first execution characteristic;

associating the certain arguments that cause the subprogram to exhibit the first execution characteristic with a portion of the subprogram;

replacing the portion of the subprogram that exhibits the first execution characteristic with program instructions that explicitly define operations of the first execution characteristic; and

leaving intact a second portion of the subprogram that exhibits the second execution characteristic.

45. (Previously Presented) The apparatus of claim 44 wherein said second execution characteristic is an atypical characteristic associated with program code that requires special processing.

46. (Original) The apparatus of claim 45 wherein said atypical characteristic is an execution time duration.

47. (Original) The apparatus of claim 46 wherein said execution time duration exceeds a predetermined threshold.

48. (Previously Presented) The apparatus of claim 47 wherein said first execution characteristic is a typical execution characteristic associated with program code that requires normal processing.

49. (Original) The apparatus of claim 47 wherein said first and second execution characteristics are execution time durations.

Claim 50. (Previously Canceled)

51. (Previously Presented) The apparatus of claim 44 wherein the certain arguments are associated with conditional execution computer statements associated with the first characteristic.

52. (Previously Presented) The apparatus of claim 47 wherein the first and second execution characteristics are invoked based on conditional execution computer statements associated with the characteristics.

53. (Currently Amended) An apparatus having a processor and a memory containing programs for replacing subprogram code in a computer system which when executed using the processor perform steps comprising, ~~comprising the steps of:~~

identifying a subprogram that operates in a first manner when operands passed to the subprogram fall within a first range of values and that operates in a second manner when operands passed to the subprogram fall within a second range of values;

associating the operands that fall within a first range of values with one of many execution paths of the subprogram; and

replacing subprogram statements included in the one execution path that cause the subprogram to operate in the first manner with expanded code.